Remarks

Status of the Claims

Claims 1-43 are pending in the application. All claims stand rejected. By this paper, claims 1, 11, 21, 31, and 41-43 have been amended. Reconsideration of all pending claims is respectfully requested.

Claim Objections

Claim 41 was objected to because of a typographical error, which has now been corrected in accordance with the Examiner's suggestions.

Claim Rejections – 35 U.S.C. § 102(a)

Claims 1-5, 9-15, 19-25, 31-35, and 39-43 were rejected under 35 U.S.C. § 102(a) as being allegedly anticipated by Beeler, Jr. ("Beeler"). This rejection is respectfully traversed.

Claims 1, 21, 41, and 42

Claim 1 has been amended to recite method for backing up a file system in a partition comprising a plurality of allocation units, comprising:

> copying each allocation unit occupied by a plurality of files of the file system to a locally-stored image file, wherein the locally-stored image file is located within the same partition as the file system being backed up; and
>
> adding a directory map to the locally-stored image file that associates copied allocation units in the locally-stored image file with names of corresponding files from the file system.

Claims 21, 41, and 42 have been variously amended to include similar limitations, and the following analysis is therefore equally applicable to them.

1. Beeler does not teach or suggest backing up a file system to a *locally-stored* image file located *within the same partition as the file system being backed up*.

FIGs. 1 and 2 of the present application illustrate the differences between the conventional approaches (as in Beeler) and the claimed invention. For example, as shown in FIG. 1 (illustrating conventional thinking), a file system 102 contained within a partition 102 (*i.e.*, Partition #1) may be backed up to an image file 110 on a separate backup storage device 107. Alternatively, the file system 102 may be backed up to an image file 110 within a different partition (*i.e.*, Partition #2). In neither case, however, is the image file 110 located within the same partition as the file system 102 being backed up, as shown in FIG. 2. Furthermore, the image files 110 of FIG. 1 cannot be referred to as being "locally" stored. Rather, they are being "remotely" stored, both remotely from Partition #1, which contains the file system being backed up, and remotely from the primary storage device 106.

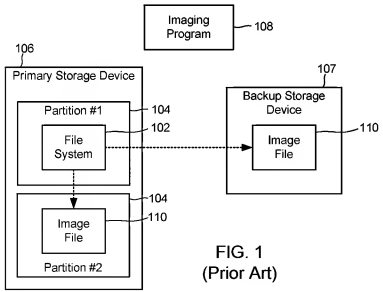


FIG. 1
(Prior Art)

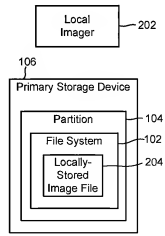


FIG. 2

As illustrated in FIG. 2, the image file 204 may actually be stored as a file within the file system 102 being backed up. In no case, however, is it stored outside of the partition of the file system 102 being backed up, either in a separate partition or a separate device.

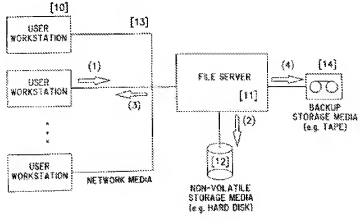
Beeler follows the conventional thinking by backing up one or more source computers to one or more target computers. As stated in paragraph [0015] of Beeler,

> The purpose of this invention is to provide means for real-time, transaction-based ***replication of one or more source computers on a network to one or more target computers***, which may or may not be running the same operating system software as the original source computer. (Emphasis added).

Furthermore, paragraph [0016] of Beeler states:

> A feature of the invention is the manner in which ***information on a computer system is replicated to a secondary storage media*** in real-time. Specifically, when a change is made to a file or configuration item on the primary (source) computer, those changes are immediately ***copied to a secondary (target) computer***. ... This ***reduces the amount of network traffic required to attain real-time replication***. (Emphasis added).

The differences between FIG. 1 of Beeler (illustrated below) and FIG. 2 of the present invention are clear. Information from one or more user workstations 10 is copied to a



file server 11 for backup on a hard disk 12 or tape 14. In neither case, however, is the image located "within the same partition as the file system being backed up," as claimed. If so, there would no need for "network traffic" or the transitions indicated by arrows 1, 2, 3, and 4.

As pointed out by the Examiner, Beeler does refer to a "Single Server mode" in connection with FIG. 6 (illustrated below), in which

> source server [61] data is replicated to local file system [63], as shown in FIG. 6. Once the data is mirrored, workstations [60] send file modification requests to source/target server. When the modification request is executed on local file system [63], the source/target server then executes the file modification request on local file system [62]. ***One skilled in the art will appreciate that local file systems [62] and [63] can be one or two non-volatile data storage device. In the case, of one storage device, the primary data and replicated data will be in <u>different volumes of the same data storage device</u>***. Further, it is always an option to attach a backup storage device to the target server.

Paragraph [0079] (emphasis added).

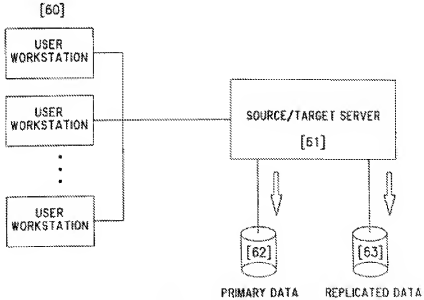


Fig. 6

As explained by Beeler, data from file system [62] may be replicated to file system [63]. However, the file systems [62] and [63] will either be stored in two different storage devices or in different volumes of the same data storage device. The terms "volume" and "partition" are often used interchangeably:

- "To create a partition or volume (the two terms are often used interchangeably) on a hard disk, there must be either unallocated (empty) space on the hard disk or free space within an extended partition on the hard disk." (http://windowshelp.microsoft.com/Windows/en-US/Help/d9a4d35e-efdf-406c-a049-0860180129a71033.mspx.)

- "In the context of computer operating systems, Volume is the term used to describe a single accessible storage area with a single filesystem, typically (though not necessarily) resident on a single partition of a hard disk... 'Logical drive' and 'volume' should be considered synonymous" (http://en.wikipedia.org/wiki/Volume_(computing)).

- "Volumes are collections of directories, subdirectories, and files." (http://www.linktionary.com/v/volume.html).

According to the specification:

> In IBM-compatible personal computers, hard drives may be divided into partitions, which are subdivisions of allocation units typically used to store a separate file system. For ***example, each partition has its own directory area, including a control data structure, volume catalog, etc.*** Accordingly, the partitions may be treated by the OS as ***separate logical storage devices.***

Specification at paragraph [0015] (emphasis added).

Thus, Beeler's reference to data being stored on different volumes means the same thing as storing the data in different partitions, which is precisely the conventional thinking illustrated in FIG. 1 of the present application. If Beeler is replicating data to a different volume, which is a "separate collection of directories, subdirectories, and files,"

he is not replicating the data to the same partition, as illustrated in FIG. 2, and recited in claim 1.

Beeler is therefore similar to FIG. 1 of the present application (illustrating conventional thinking), in which a file system on a primary storage device (source computer) is backed up to an image file on a backup storage device (target computer), or, as in the case of Beeler's FIG. 6, backed up to a different volume/partition. Beeler does not teach or suggest creating a locally-stored image within the same partition, as claimed.

2. Beeler does not teach or suggest creating ***a*** locally-stored image ***file***.

It is unclear whether Beeler actually creates a single image file representing the backed up file system, as claimed, or simply copies files from one location to another. The Office Action argues in favor of the latter interpretation, contrary to the claim language. According to the Office Action, "***directories and files*** are copied to a ***separate directory*** which would include a directory map to access the files." Office Action at page 2 (emphasis added). If this is the case, then Beeler is not creating a single locally-stored image file, but is rather copying multiple files between locations.

3. Beeler teaches away from copying ***each*** allocation unit occupied by a plurality of files of the file system to a locally-stored image file.

Claim 1 recites "copying *each* allocation unit occupied by a plurality of files of the file system to a locally-stored image file." However, Beeler requires that

> [o]nly data that has been changed on the source computer is transmitted to the target computer for replication, versus transmitting the entire contents of the file.

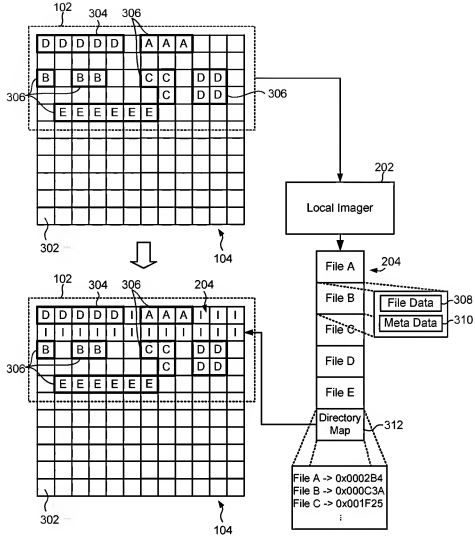
Beeler at [0016]. Beeler cites the advantages of "reduce[ing] the amount of network traffic required to attain real-time replication." *Id.* However, this teaches away from copying each allocation unit occupied by a file, since Beeler is not "transmitting the entire contents of the file."

4. Beeler does not teach or suggest adding a directory map adding a directory map to the locally-stored image file that associates copied allocation units in the locally-stored image file with names of corresponding files from the file system.

The Office Action appears to be arguing that files are being copied to a separate directory, and that a directory map is therefore being created to provide access to the files. Office Action at page 2. However, the "directory map" being referred to by the Examiner is the directory area 304 (illustrated in FIG. 3) maintained by the operating system (OS), not a separate directory map 312 being added to a locally-stored image file 204.

As noted above, it is unclear whether Beeler even creates a locally-stored image file. In fact, the Office Action seems to contradict this by stating that "***directories and files*** are copied to a ***separate directory*** which would include a directory map to access the files." Office Action at page 2 (emphasis added).

As illustrated below in FIG. 3 of the present application, a local imager 202 creates an image file 204 containing all of the file data 308 and meta data 310 from the file system 102. The directory map 312 is added to link file names to the stored file data 308 and meta data 310 within the image file 204. However, the directory map 312 exists separate from, and in addition to, the directory area 304 maintained by the OS, which is what the Examiner is referring to at page 2 of the Office Action.



Beeler is silent about adding a separate directory map 312 to a locally-stored image file 204. Indeed, as noted above, Beeler appears to teach away from creating an image file 204, as opposed to simply copying files, as suggested by the Examiner.

Anticipation under section 102 is proper only if the reference shows exactly what is claimed. Titanium Metals Corp. v. Banner, 778 F.2d 775, 780 (Fed. Cir. 1985); MPEP § 2131. Based on the foregoing, is clear that Beeler is deficient is several respects. Beeler does not teach or suggest backing up a file system to an *locally-stored* image file

within the same partition as the file system being backed up. In fact, Beeler teaches away from copying *each* allocation unit occupied by a plurality of files of the file system to a locally-stored image file. Finally, Beeler does not teach or suggest adding a directory map adding a directory map to the locally-stored image file that associates copied allocation units in the locally-stored image file with names of corresponding files from the file system. The other cited references (Milligan and Hastings) do not teach or suggest any of these elements. Accordingly, the section 102 rejection is improper and should be withdrawn.

Claims 11, 31, and 43

Claims 11, 31, and 43 variously recite the process of restoring a file system to a partition, including the limitation of:

> accessing a locally-stored image file located within the partition to which the file system is to be restored, the locally-stored image file comprising a directory map and file data for a plurality of files.

As discussed above, Beeler does not disclose or suggest locating the image file within the file system to be backed up. Beeler is similar to FIG. 1 of the present application (illustrating conventional thinking), in which a file system on a primary storage device (source computer) is backed up to an image file on a backup storage device (target computer), or, as in the case of Beeler's FIG. 6, backed up to a different volume/partition. Beeler does not teach or suggest creating a locally-stored image within the same partition, as claimed. Since no locally-stored image is created within the same partition, such an image will not be within the partition when it is time for the file system contained therein to be restored. In Beeler, the image file (if one is actually

created) would be stored within a separate target computer or in a different volume/partition.

Milligan and Hastings do not teach or suggest any of these elements. Accordingly, claims 11, 31, and 43 contain patentably subject matter, and the section 102 rejection should be withdrawn.

Claims 9, 10, 19, 20

Claims 9 and 19 recite protecting the locally-stored image file from accidental deletion or modification. Claims 10 and 20 recite two different protection methods:

- providing a filter driver that intercepts and denies requests to access the locally-stored image file; and
- initiating a process that opens and thereby locks the locally-stored image file.

Because the image file is stored within the same partition as the file system being backed up (rather than a separate partition/volume or different backup device), it is more vulnerable to being inadvertently deleted or modified. Accordingly, various techniques are provided to reduce the likelihood of such an occurrence.

According to the Office Action, claim 9 is taught in paragraph [0112] of Beeler, the relevant portion of which is reproduced below for the Examiner's convenience.

> The source server [250] may only send a limited number of mirror packets [258] at a time, in order to prevent locking out replication and other applications from network resources.

Not only does the cited paragraph not teach protecting an image file from deletion or modification, it appears to teach the opposite of what is being claimed, *i.e.*, preventing

locking out replication and other applications from network resources. The phrase "***locking out*** replication and other applications from network resources" is being modified by "***preventing***," which implies that replication and access to network resources is being allowed.

Claims 10 and 20 recite "initiating a process that opens and thereby locks the locally-stored image file." Preventing "locks" on the locally-stored image file, as described by Beeler, is directly contrary to what is being claimed. Accordingly, the section 102 rejection of claims 9, 10, 19, and 20 should be withdrawn.

Even if Beeler somehow taught the concept of locking the image file (which it does not), Beeler does not disclose or suggest the specific locking mechanisms recited in claims 10 and 20. Beeler does not disclose or suggest a filter driver that intercepts and denies requests to access the locally-stored image file. Similarly, Beeler does not suggest initiating a process that opens and thereby locks the locally-stored image file. These specific locking mechanisms cannot be derived from a generic discussion of file locking, even if one were included in the cited references.

Claim Rejections – 35 U.S.C. § 103(a)

Claims 6-8, 16-17, 26-30, and 36-37 were rejected under 35 U.S.C. 103(a) as being allegedly unpatentable over Beeler in view of Milligan et al. ("Miligan"). Claims 18 and 38 were rejected under 35 U.S.C. 103(a) as being allegedly unpatentable over Beeler in view of Hastings. For the reasons explained below, these rejections are respectfully traversed.

Claims 6-8, 16-17, 26-30, and 36-37

Claim 6 recites "marking a beginning point of the locally-stored image file to assist in locating the locally-stored image file in the event of directory area corruption." The directory area 304 (shown in FIG. 3 of the present application) provides a mechanism for locating files stored in the partition including the locally-stored image file 204. However, if the directory area 304 becomes corrupted, it may be difficult or impossible to distinguish the locally-stored image file 204 from other file data. Marking the beginning point of the locally-stored image file with a unique "beginning-of-image marker" (claim 7) allows the system to sequentially look through the file data to find the image file 204. If the image file 204 has been defragmented (claims 18 and 38), the system can more easily locate the entire image file 204 and thereby restore the file system 102 and repair the corrupted directory area 304.

According to the Office Action at page 12, Beeler does not explicitly teach this limitation. Applicants agree. However, the addition of Milligan does not cure the deficiencies of Beeler.

The Office Action argues that Milligan teaches this limitation at Figure 4, which, according to the Examiner, "illustrates a pointer to a track level where a record is stored to provide structure level points that are not fixed and can achieve fine granularity without requiring enormous number of pointers." Applicants are unsure how the pointers discussed in Milligan are at all relevant to marking the beginning point of the locally-stored image file to assist an image restoration program in locating the locally-stored image file in the event of directory area corruption, let alone storing a unique

beginning-of-image marker at an initial allocation unit occupied by the locally-stored image file. Milligan has nothing to do with backing up file systems. Rather, Milligan relates to the problem of "performing a data file copy operation in a manner that minimizes the use of processing resources and data storage memory." Paragraph [0005].

According to Milligan,

> the pointer systems used in known instant copy operations make use of fixed level pointers. That is, as shown in FIG. 4, the pointer storage structure 450, e.g., a pointer table, includes a plurality of pointers. The pointers may point to a storage volume 410, a cylinder 420, or a track 430 level of the data storage device. However, the level to which the pointer points is fixed. In the depicted example, pointer 1 points to a storage volume 410 level, pointer 2 points to a cylinder level 420, and pointer 3 points to a track level 430 of the data storage device.

Milligan at [0029].

Creating a separate pointer storage structure 450, which points to tracks, cylinders, or storage volumes, is very different from storing a unique beginning-of-image marker in the initial allocation unit of the image file to mark its beginning point. If Milligan's pointer storage structure 450 became corrupted, how would an image restoration program be able locate the tracks, cylinders, or volumes to which it was pointing? Claims 6-8 deal with just that eventuality. If the directory area (or pointer structure) is corrupted, some kind of marker needs to be stored in the data to identify where the pointed-to file begins. Miligan does not teach or suggest this, but seems to teach away from it by relying on an external pointer structure.

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. MPEP § 2143.03. As

demonstrated above, Milligan does not teach or suggest all of the deficiencies of Beeler. Accordingly, the section 103 rejection should be withdrawn.

Claims 18 and 38

Claims 18 and 38 variously recite "defragmenting the locally-stored image file within the partition prior to extracting the file data." As shown in FIG. 8 of the present application, after initializing the allocation units 302 and prior to extracting the file data 308, an image defragmenter 802 may defragment the locally-stored image file 204. Defragmentation is the process of reconfiguring the allocation units 302 used by a file such that the file is entirely stored along a logically-sequential series of discrete allocation units 302.

Once the locally-stored image file 204 has been defragmented, the restoration process may proceed, as described in FIG. 7, to extract the file data 308 and create the new directory area 304. When complete, any fragmentation that existed in the original file system 102 will have been eliminated ensuring optimal file access times.

The Office Action refers to paragraph 46 of Hastings for the proposition that "a file system should first be defragmented (if corrupted) as a first-level solution." Applicants respectfully notes that there is no paragraph 46 in Hastings, and, furthermore, cannot find the words "defragmented or "corrupted" in the reference. However, even assuming that a reference taught what was recited in the rejection, Applicants respectfully submit that this is not what is being claimed. Applicant is not claiming the process of defragmenting a corrupted file system. Indeed, the original file system is initialized (deleted) before the files are extracted from the locally-stored

image. See paragraph [0068] of the present application. As illustrated below in FIG. 8 of the present application, only the image file 204 is defragmented, so that once the files are extracted from it, they will automatically be defragmented.

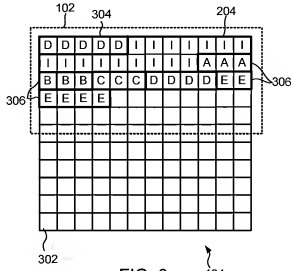


FIG. 8

Defragmentation of the image file prior to extraction of the image data is not taught or suggested by Hastings or any of the other art of record. Accordingly, the section 103 rejection should be withdrawn.

Conclusion

In view of the foregoing, all pending claims are believed to be allowable. A Notice of Allowance is respectfully requested. If any impediment remains to the prompt allowance of the above-identified application, Applicant respectfully requests that the Examiner contact the undersigned at the telephone number listed below

Respectfully submitted,

/Kory D. Christensen/
Kory D. Christensen
Registration No. 43,548

STOEL RIVES LLP
One Utah Center Suite 1100
201 S Main Street
Salt Lake City, UT 84111-4904
Telephone: (801) 328-3131
Facsimile: (801) 578-6999